

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

- 5 5. A method for formatting a document, comprising the steps of:
receiving a user example; and
formatting the non-functional aspects of said document in the style of said user example.
6. The method of claim 1 wherein said non-functional aspects include indentation, order, and comment style.
7. A method for formatting an output document, comprising the steps of:
receiving from a user an example document;
selectively generating from said example document style templates, alignment offsets and section order; and
responsive to said templates, offsets and order, formatting functional aspects of said output document.
8. A method for generating an output document in a user preferred style, comprising the steps of:
capturing the user preferred style from a user example document; and
generating a plurality of templates, each said template representing a component of said user example document and selectively including replaceable macros.
- 20 9. The method of claim 4, further comprising the steps of:
generating functional aspects;
replacing said macros in said template with information from said functional aspects; and
responsive to said template with information from said functional aspects, generating said output document.

6. The method of claim 4 or claim 5, further comprising the step of applying syntactical patterns to said user example document to define said component.

7. The method of claim 6, further comprising the step of temporarily removing comments from said user example document when applying said syntactical patterns to said user example document.

5 8. The method of any one of claims 4 to 7, said output document including a plurality of separately generated sections.

9. The method of any one of claims 4 to 8, further comprising the steps of:
determining from said user example document a user preference for group order; and
generating said output document with public, protected, and private member access in said user preference order.

10. The method of any one of claims 4 to 9, further comprising the step of receiving from said user further input changing the style of said user example document.

11. The method of any one of claims 4 to 10, wherein said replaceable macros correspond to text in said user example document.

12. A method for generating an output document with indentation of document components in a user preferred style, comprising the steps of:

receiving a user example document;

while parsing document components in said user example document, preserving the relative indentation of subcomponents by calculating user desired offsets for said subcomponents; and

responsive to said user desired offsets, generating said output document.

13. The method of claim 12, said user desired offsets being preserved for variables, functions, and constructors.

14. A method for generating an output document in a user preferred style, comprising the steps of:

reading an example file representing said user preferred style into an input buffer;

searching said input buffer for a pattern that matches that of an expected section;

if said pattern is found,

from the position of said pattern, defining a first bound by searching backwards in said buffer until a previous expected search pattern is found;

from the position of said pattern, defining a second bound by searching forwards in said buffer until a next expected search pattern is found;

copying a string of characters contained within said input buffer between said first bound and said second bound to a template buffer;

parsing said template buffer to isolate expected keywords, and names and subsections;

if said expected section is a section that can be repeated in a document, saving in said template buffer the line offsets of keywords, names and other elements;

replacing content-specific subsections with macro names; and

if said pattern is not found,

creating a default template buffer for said expected section.

15. The method of claim 14, comprising the further steps of:

getting a said template buffer for each section to be generated in said output document;

getting user content for all sections of said output document;

creating an output buffer for storing said output document;

for each section of said output document,

putting a corresponding template buffer into a temporary output buffer;

replacing macro names in said temporary output buffer with user content information;

if this section is expected to be repeated and the user desires alignment, using corresponding template offsets to modify said temporary output buffer for aligning keywords, names, and other sub-sections;

inserting the content of said temporary output buffer into said output buffer; and
writing said output buffer to a file.

16. A computer program product for generating an output document in a user preferred style, said computer program product comprising:

5 a style capture tool for examining an input document containing an example of said user preferred style to determine said user preferred style for non-functional aspects of said output document;

a code generation tool for generating functional aspects of said output document; and

10 a document generate tool responsive to said style capture tool and said code generation tool for generating said output document with said preferred style for non-functional aspects applied to the presentation of said functional aspects.

17. The computer program product of claim 16, further comprising:

a grammar template for storing syntax rules; and

15 said style capture tool being responsive to said syntax rules for pattern-matching said user example document.

18. The computer program product of claim 17, further comprising a plurality of grammar templates, each said template for storing syntax rules for a unique one of a plurality of programming languages.

19. The computer program product of any one of claims 16 to 18, further comprising a plurality
20 of input document files, each said input document file representing a user preferred style for different parts of said output document.

20. The computer program product of claim 19, said input document files including a declaration example file and a definition example file.

21. The computer program product of any one of claims 16 to 20, wherein said code generation tool is operable for generating class declarations, and said style capture tool is operable for providing to said document generate tool rules for syntax and ordering of class-head, base-specifiers, class body, access-specifiers, and member-declarations.

5 22. A computer program product for generating an output document, said computer program product comprising:

at least one grammar template file, one said grammar template file for each of one or more sections of an output document in one or more programming languages, each said grammar template file for specifying the manner for parsing and defining the bounds of a section of said output document; and

10 at least one style template parsed from a user example document in a user preferred style using said grammar template file for defining the style of a section of said output document.

23. The computer program product of claim 22, further comprising:

15 a syntax template for finding and extracting style information for each section of said user example document and including a section identifier, an external pattern, an internal pattern, a before pattern, an after pattern, a repeatability indicator, and an ordering indicator; each said syntax template being associated with a single style template;

said section identifier for identifying a section of said output document;

said external pattern for finding a particular section in said input document;

20 said internal pattern for indicating the textual elements to be considered as part of said particular section;

said before pattern for indicating what should come before said particular section;

said after pattern for indicating what should come after said particular section;

25 said repeatability indicator for indicating whether said particular section is a repeatable section and, if so, that alignment offsets need to be calculated; and

said ordering indicator for indicating if said particular section is part of a group of unique sections and, if so, whether the ordering of said group of unique sections is independent or whether

the ordering of said group must be captured from said user example document.

24. A computer program product for formatting a document, said computer program product comprising:

instruction means for receiving a user example; and

5 instruction means for formatting the non-functional aspects of said document in the style of said user example.

25. The computer program product of claim 24 wherein said non-functional aspects include indentation, order, and comment style.

26. A computer program product for formatting documents, said computer program product comprising:

instruction means for receiving from a user an example document;

instruction means for selectively generating from said example document style templates, alignment offsets and section order; and

instruction means for, responsive to said templates, offsets and order, formatting functional aspects of said output document.

27. A computer program product for generating an output document in a user preferred style, said computer program product comprising:

instruction means for capturing the user preferred style from a user example document; and

20 instruction means for generating a plurality of templates, each said template representing a component of said user example document and selectively including replaceable macros.

28. The computer program product of claim 27, further comprising:

instruction means for generating functional aspects;

instruction means for replacing said macros in said template with information from said functional aspects; and

instruction means for, responsive to said template with information from said functional aspects, generating said output document.

29. The computer program product of claim 27 or claim 28, further comprising instruction means for applying syntactical patterns to said user example document to define said component.

30. The computer program product of claim 29, further comprising the step of instruction means for temporarily removing comments from said user example document when applying said syntactical patterns to said user example document.

31. The computer program product of any one of claims 27 to 30, said output document including a plurality of separately generated sections.

32. The computer program product of any one of claims 27 to 31, further comprising:
instruction means for determining from said user example document a user preference for group order; and
instruction means for generating said output document with public, protected, and private member access in said user preference order.

33. The computer program product of any one of claims 27 to 32, further comprising instruction means for receiving from said user further input changing the style of said user example document.

34. The computer program product of any one of claims 27 to 33, wherein said replaceable macros correspond to text in said user example document.

35. A computer program product for generating an output document with indentation of document components in a user preferred style, said computer program product comprising:
instruction means for receiving a user example document;
instruction means for, while parsing document components in said user example document,

preserving the relative indentation of subcomponents by calculating user desired offsets for said subcomponents; and

instruction means for, responsive to said user desired offsets, generating said output document.

36. The computer program product of claim 35, said user desired offsets being preserved for variables, functions, and constructors.

37. A computer program product for generating an output document in a user preferred style, said computer program product comprising:

instruction means for reading an example file representing said user preferred style into an input buffer;

instruction means for searching said input buffer for a pattern that matches that of an expected section;

if said pattern is found,

instruction means for, from the position of said pattern, defining a first bound by searching backwards in said buffer until a previous expected search pattern is found;

instruction means for, from the position of said pattern, defining a second bound by searching forwards in said buffer until a next expected search pattern is found;

instruction means for copying a string of characters contained within said input buffer between said first bound and said second bound to a template buffer;

instruction means for removing said string from said input buffer;

instruction means for parsing said template buffer to isolate expected keywords, and names and subsections;

instruction means for, if said expected section is a section that can be repeated in a document, saving in said template buffer the line offsets of keywords, names and other elements;

instruction means for replacing content-specific subsections with macro names; and

if said pattern is not found,

instruction means for creating a default template buffer for said expected section.

38. The computer program product of claim 37, further comprising:

instruction means for getting a said template buffer for each section to be generated in said output document;

instruction means for getting user content for all sections of said output document;

instruction means for creating an output buffer for storing said output document;

for each section of said output document,

instruction means for putting a corresponding template buffer into a temporary output buffer;

instruction means for replacing macro names in said temporary output buffer with user content information;

instruction means for, if this section is expected to be repeated and the user desires alignment, using corresponding template offsets to modify said temporary output buffer for aligning keywords, names, and other sub-sections;

instruction means for inserting the content of said temporary output buffer into said output buffer; and

instruction means for writing said output buffer to a file.

39. A program storage device readable by a machine, tangibly embodying a program of instructions executable by a machine to perform the method steps of any one of claims 1 to 15.

A system for generating an output document in a user preferred style, comprising:

a style capture tool for examining an input document containing an example of said user preferred style to determine said user preferred style for non-functional aspects of said output document;

a code generation tool for generating functional aspects of said output document; and

a document generate tool responsive to said style capture tool and said code generation tool

for generating said output document with said preferred style for non-functional aspects applied to the presentation of said functional aspects.

41. A system for generating an output document, comprising:

at least one grammar template file, one said grammar template file for each of one or more sections of an output document in one or more programming languages, each said grammar template file for specifying the manner for parsing and defining the bounds of a section of said output document; and

at least one style template parsed from a user example document in a user preferred style using said grammar template file for defining the style of a section of said output document.

42. A system for generating an output document in a user preferred style, comprising:

means for capturing the user preferred style from a user example document; and

means for generating a plurality of templates, each said template representing a component of said user example document and selectively including replaceable macros.

43. The system of claim 42, further comprising:

means for generating functional aspects;

means for replacing said macros in said template with information from said functional aspects; and

means for, responsive to said template with information from said functional aspects, generating said output document.